



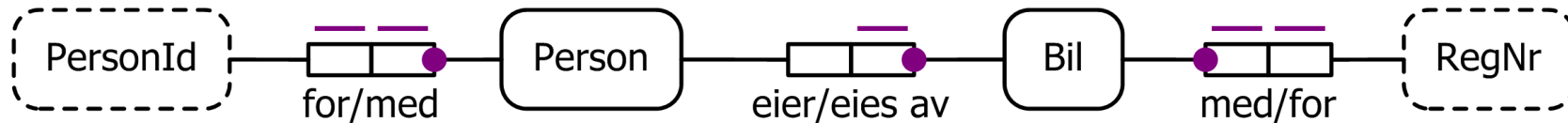
INF1300

Introduksjon til databaser

Dagens tema:

Realiseringsalgoritmen
(også kalt "grupperingsalgoritmen") –
fra ORM-diagram til relasjonsskjema

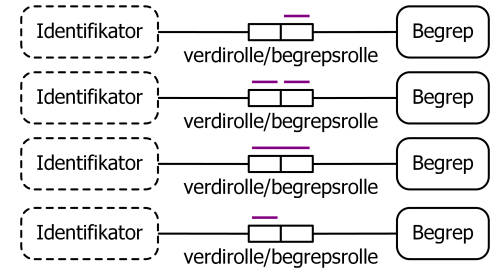
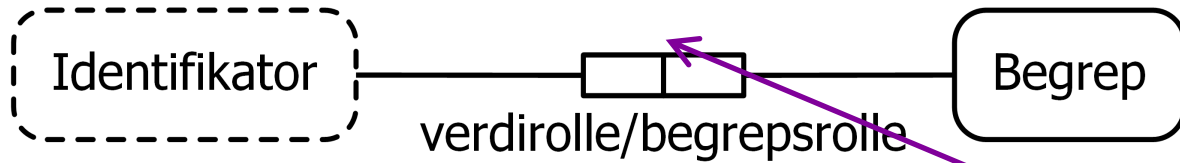
Underliggende idé (forenklet)



- For hvert **begrep**: lag en tabell
 - For hver **faktatype**: lag en tabell
 - **Perfekte broer** brukes til å bestemme hvordan begrepene skal representeres
 - **Entydigheter** brukes til å bestemme primærnøkler i tabellene
 - For å få en "penere" database: slå sammen/**grupper** tabeller med samme primærnøkkel
- Person(), Bil()
 - eier/eies_av(,)
 - Person(PersonId)
Bil(RegNr)
eier/eies_av(PersonId, RegNr)
 - Person(PersonId)
Bil(RegNr)
eier/eies_av(PersonId, RegNr)
 - Person(PersonId)
Bil(RegNr, PersonId)

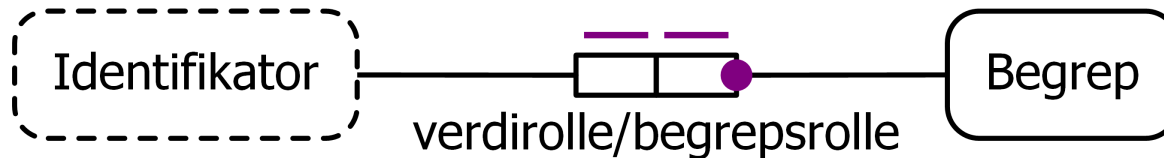
Setningstyper

- Bro

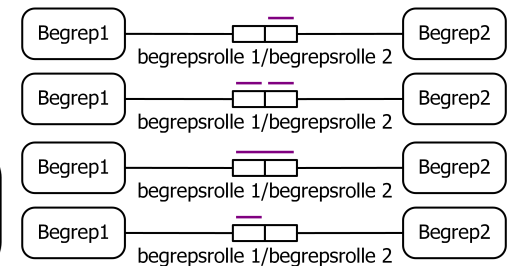
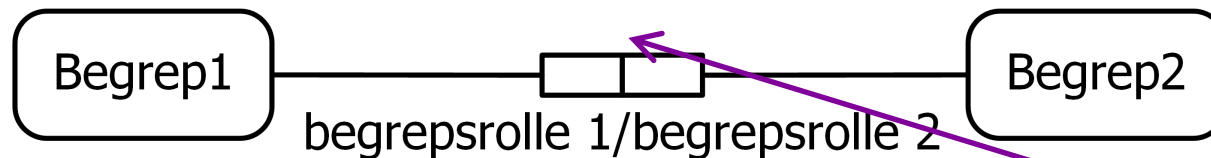


Minst én entydighetskranke

- Perfekt bro



- Faktatype
(vi viser bare en **binær** faktatype her)

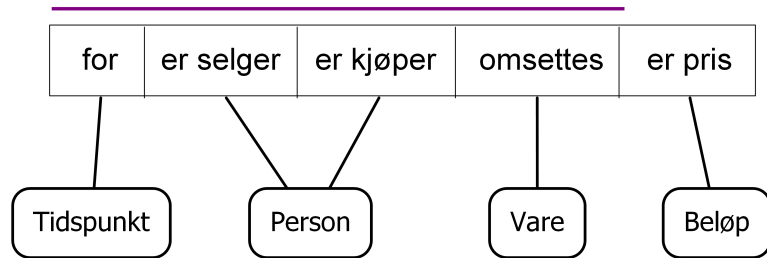


Minst én entydighetskranke

Forutsetninger/forberedelser

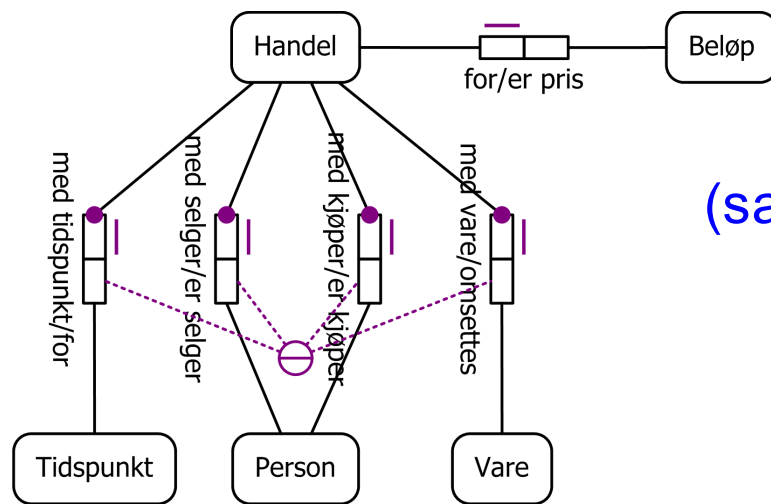
- A. Alle **lange entydighetsskranger** er omformet til korte ved begrepsdannelse
- B. ORM-diagrammet må være **refererbart**
- C. Diagrammet må ikke inneholde **synonyme broer**: alle broer må ha en entydig begrepsrolle

A. Begrepsdannelse av lange entydighetskranker

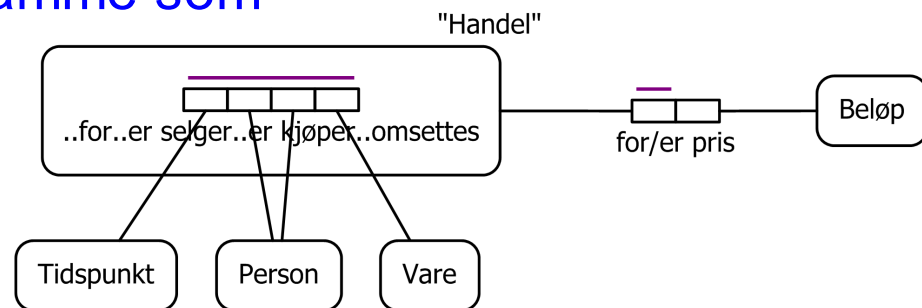


En lang entydighet er en ekstern entydighet i forkledning

erstattes av



(samme som

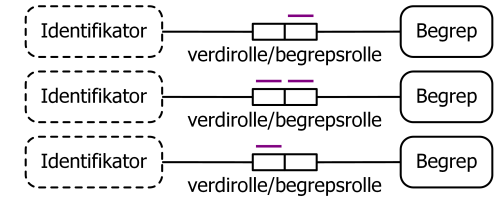
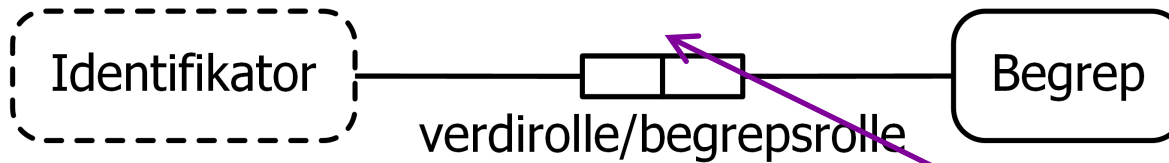


B. Refererbare ORM-diagrammer

- Intuitivt er et ORM-diagram **refererbart** hvis alle begreper kan representeres entydig (via perfekte broer)
- Vi kommer tilbake til hva som skal til for å gjøre et ORM-diagram refererbart

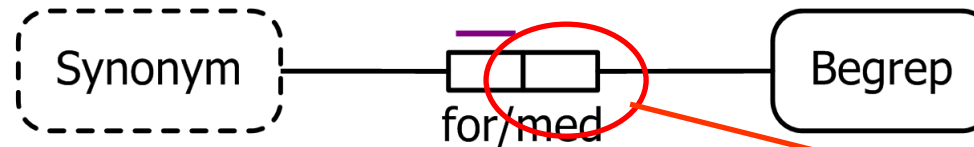
C. Eliminasjon av synonyme broer

Bro:



Etter begrepsdannelsene:
alltid én eller to **korte**
entydighetskranker

Synonym
bro:



erstattes av

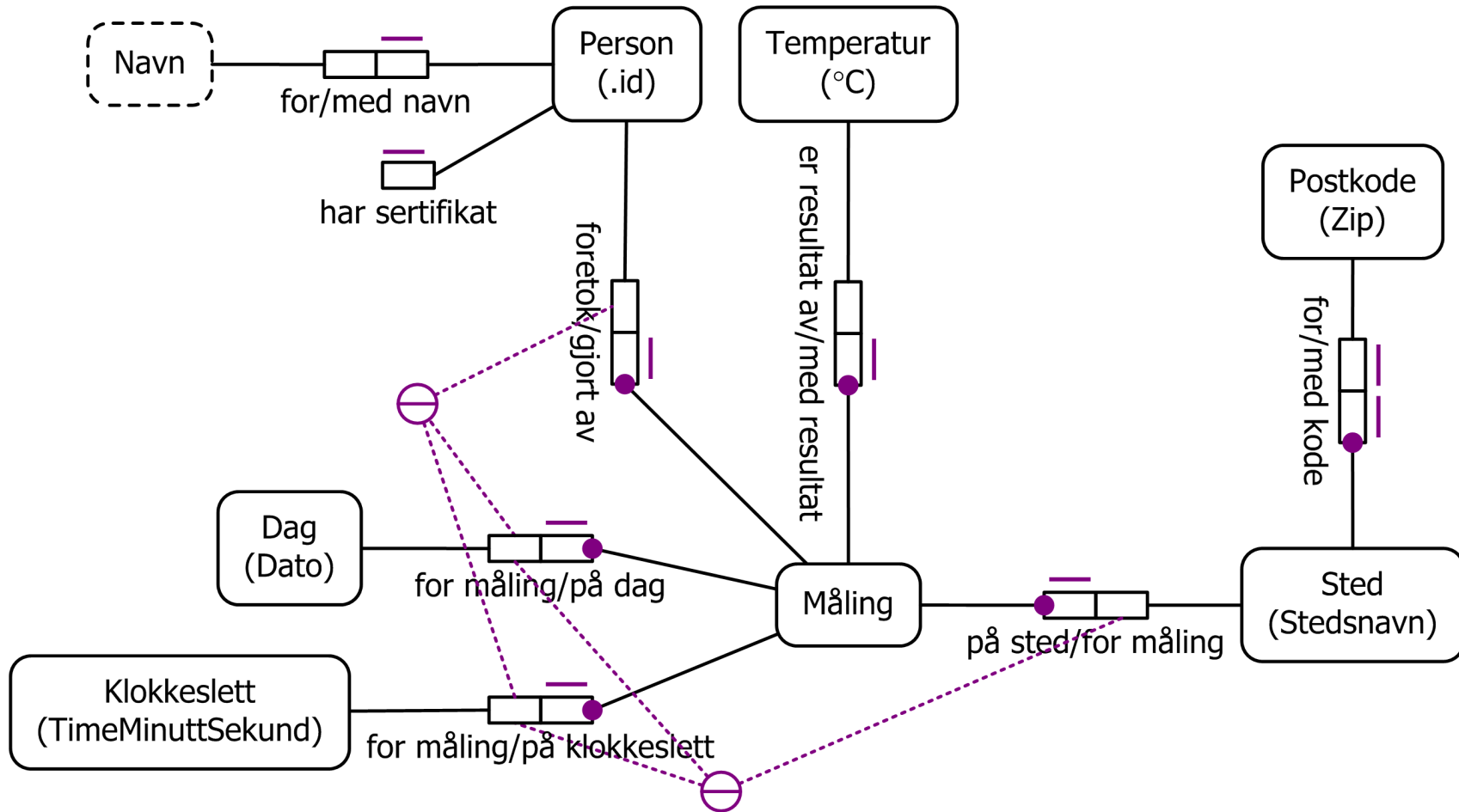
entydighetskranke
mangler på
begrepsrollen



Realiseringsalgoritmen

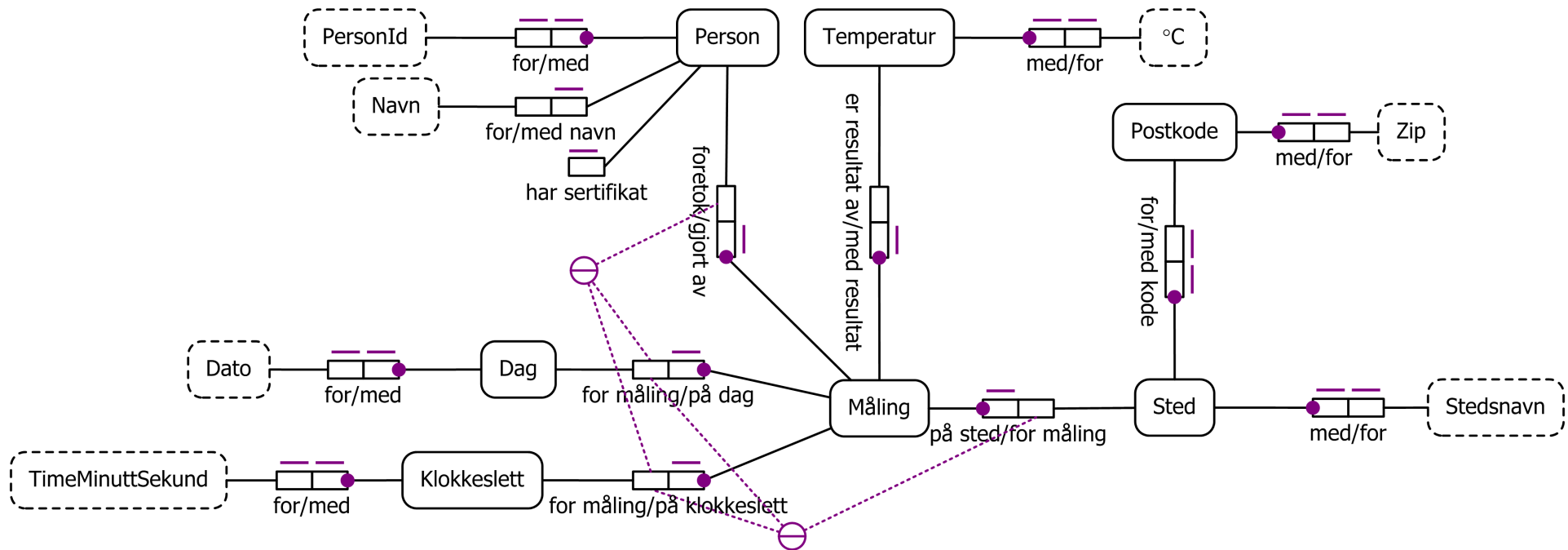
1. Hvert begrep gir opphav til en relasjon (tabell) med samme navn som begrepet
2. Finn referansemåte for alle begreper og marker alle broer og faktatyper som er benyttet til dette, som brukt
(Referansemåtene blir primærnøkler)
3. Grupper resterende broer til sine respektive begreper
(Hver bro gir ett attributt i tabellen)
4. Grupper resterende faktatyper
(Hver faktatype blir en fremmednøkkel)
5. Overfør resten av skrankene til relasjonsskjemaet
6. Bestem hvilke referanserelasjoner som skal fjernes

Eksempel



Steg 1: Fra begrep til tabell

1. Hvert begrep gir opphav til en relasjon (tabell) med samme navn som begrepet



Relasjoner: **Person, Dag, Klokkeslett, Postkode, Sted, Måling, Temperatur**

Steg 2: Valg av referansemåter

2. Finn referansemåte for alle begreper og marker alle broer og faktatyper som er benyttet til dette, som brukt (Referansemåtene blir primærnøkler)

- **Referansemåten** til et begrep er definert på neste side
- Hvis et begrep mangler referansemåte, er ORM-diagrammet ikke refererbart og kan ikke realiseres!

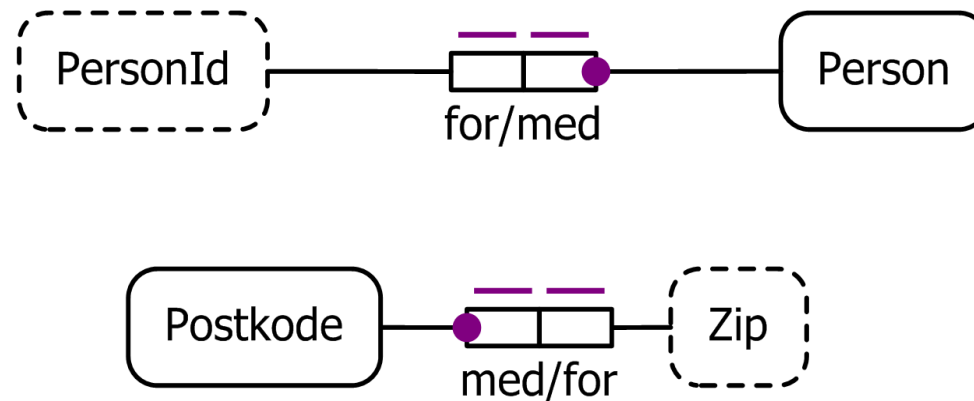
Referansemåten til et begrep

Referansemåten til et begrep er

- enten: navnet på en identifikator knyttet til begrepet med en perfekt bro
- eller: referansemåten til et annet begrep som er knyttet til begrepet via en 1:1-faktatype med påkrevd rolle
- eller: for begrepsdannelser, samlingen av referansemåtene til de begrepene som utgjør grunnlaget for begrepsdannelsen

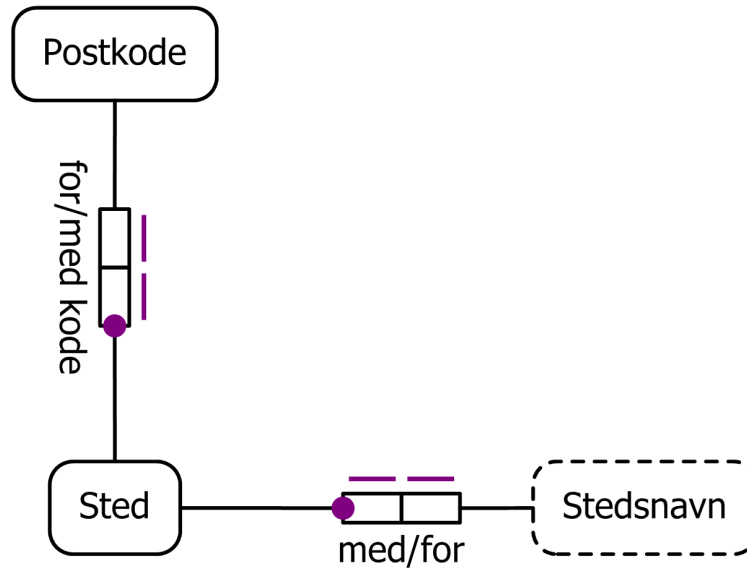
(Detaljer følger)

Referanseemåte via perfekt bro



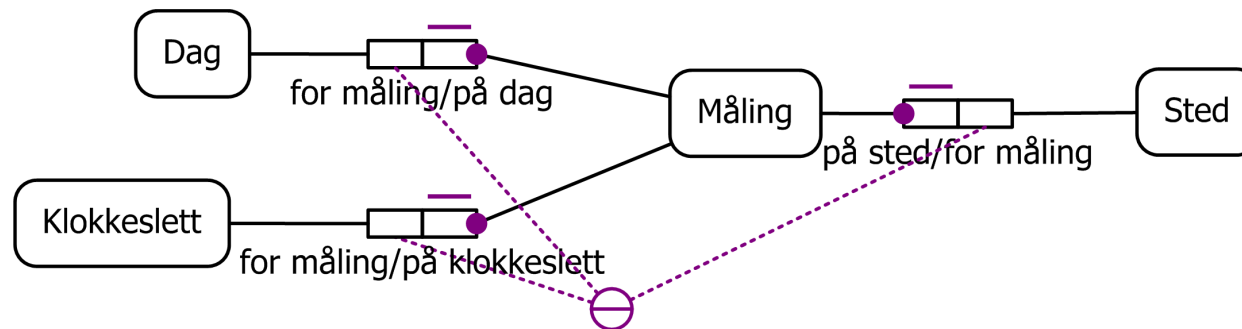
- Referanseemåten til **Person** er **PersonId**
- Referanseemåten til **Postkode** er **Zip**

Referansemåte via 1:1-faktatype



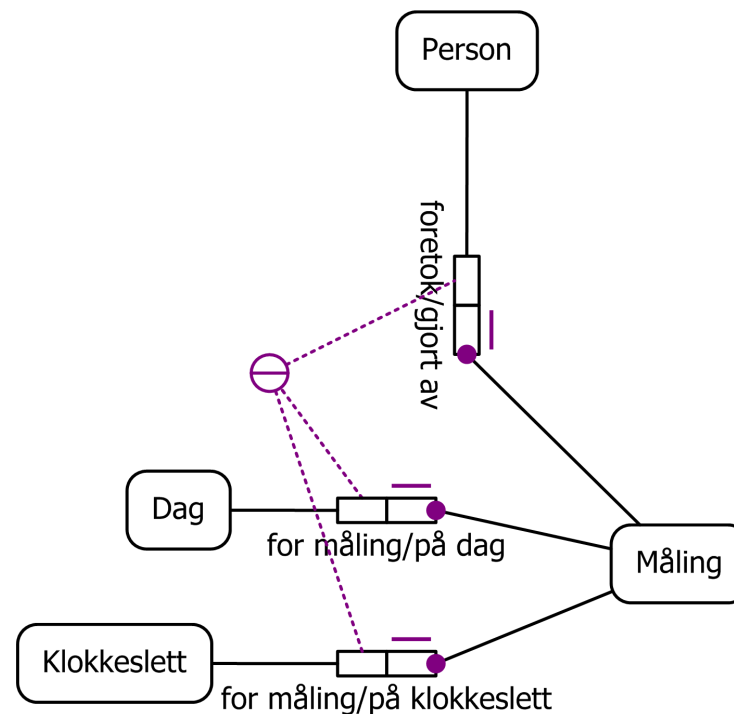
- For begrepet **Sted** har vi to mulige referansemåter:
 - Via en perfekt bro til **Stedsnavn**
 - Via en 1:1-faktatype med påkrevd rolle til **Postkode**
- Vi velger f.eks. den perfekte broen; referansemåten til **Sted** blir da **Stedsnavn**
- Alternativt kan vi velge 1:1-faktatypen; da arver **Sted** referansemåte fra **Postkode** og får referansemåten **Zip**

Referanseemåte via begrepsdannelse



- Begrepet **Måling** er en begrepsdannelse basert på begrepene **Dag**, **Klokkeslett** og **Sted**:
 1. **Måling** har entydige påkrevde roller **på dag**, **på klokkeslett**, **på sted**
 2. Det er en ekstern entydighetsskranke over rollene til **Dag**, **Klokkeslett** og **Sted**
- Referanseemåten til **Dag** er **Dato**
Referanseemåten til **Klokkeslett** er **TimeMinuttSekund**
Referanseemåten til **Sted** er valgt til **Stedsnavn**
En referanseemåte til **Måling** er derfor
(**Dato**, **TimeMinuttSekund**, **Stedsnavn**)

Referansemåte via begrepsdannelse



Alternativt kunne vi valgt referansemåte til **Måling** basert på begrepene **Dag**, **Klokkeslett** og **Person**; syntaktisk er det ikke noe i veien for dette, men begrepsmessig er det ikke like naturlig

Referansemåtene blir primærnøkler

Referansemåtene blir **primærnøkler** i de tilhørende relasjonene:

Person(PersonId)

Dag(Dato)

Klokkeslett(TimeMinuttSekund)

Postkode(Zip)

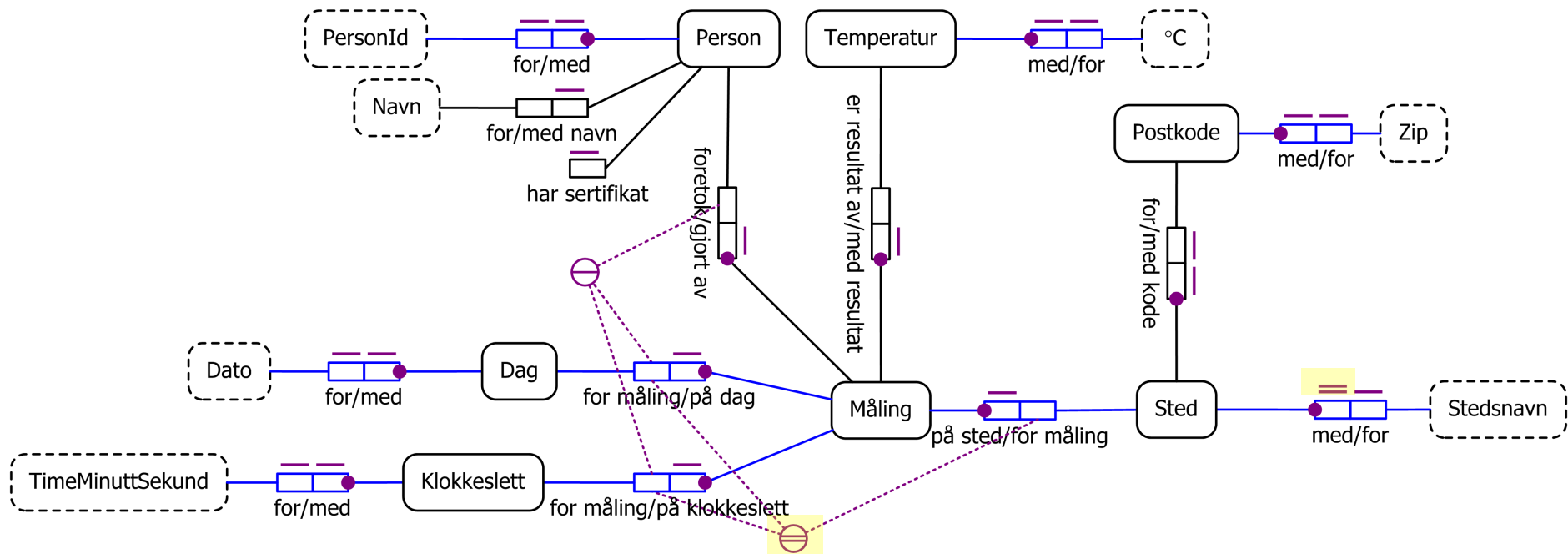
Sted(Stedsnavn)

Måling(Dato, TimeMinuttSekund, Stedsnavn)

Temperatur(°C)

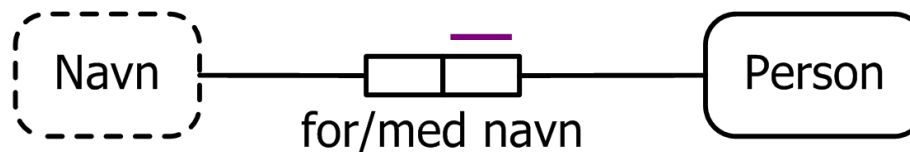
Status etter steg 2

- Alle gjenværende faktatyper er unære eller binære og har minst én kort entydighetsskranke
- Alle gjenværende broer har kort entydighetsskranke på begrepsrollen
- I ORM kan man, hvis det er flere alternativer, angi valg av referansemåte med dobbel entydighetsskranke. Under er i tillegg de brukte setningstypene markert med blått (de gjenværende er fortsatt svarte):



Steg 3: Gruppering av broer

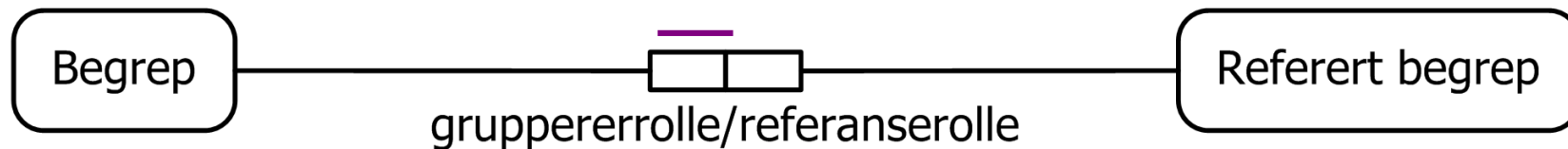
3. Grupper resterende broer til sine respektive begreper (Hver bro gir ett attributt i tabellen)



- Relasjonen **Person** får attributtet **Navn_for**:
Person(PersonId, Navn_for)
- Hvis en begrepsrolle er påkrevd, er det ikke tillatt med nullverdier i det tilhørende attributtet
- Her er begrepsrollen (**med navn**) ikke påkrevd
- Vi kan markere at nullverdier er tillatt ved å sette klammer rundt attributtet:
Person(PersonId, [Navn_for])

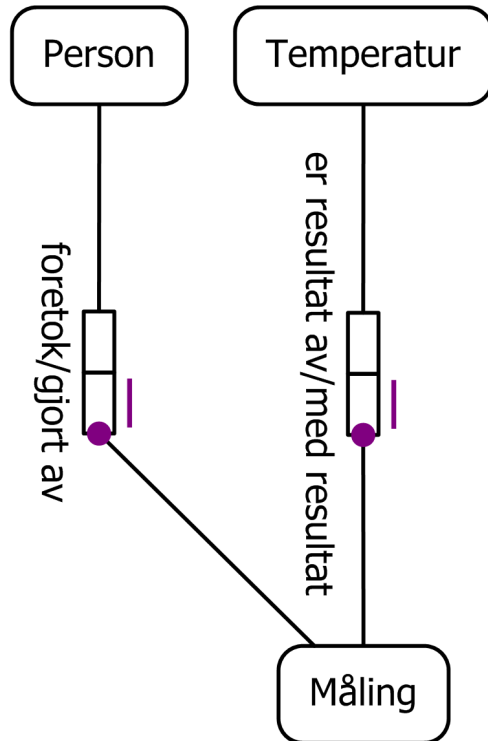
Steg 4: Gruppering av faktatyper

4. Grupper resterende faktatyper (Hver faktatype blir en fremmednøkkel)



- I binære faktatyper velges en **entydig** rolle som **gruppererrolle**; den andre kalles **referanserollen**
 - Hvis begge rollene er entydige og en av dem er påkrevd, velges den påkrevde rollen som gruppererrolle
- Relasjonen til gruppererrollens begrep får en fremmednøkkel til det refererte begrepets relasjon
- Hvis gruppererrollen er påkrevd, er det ikke tillatt med nullverdier i fremmednøkkelattributtene

Gruppering av binære faktatyper

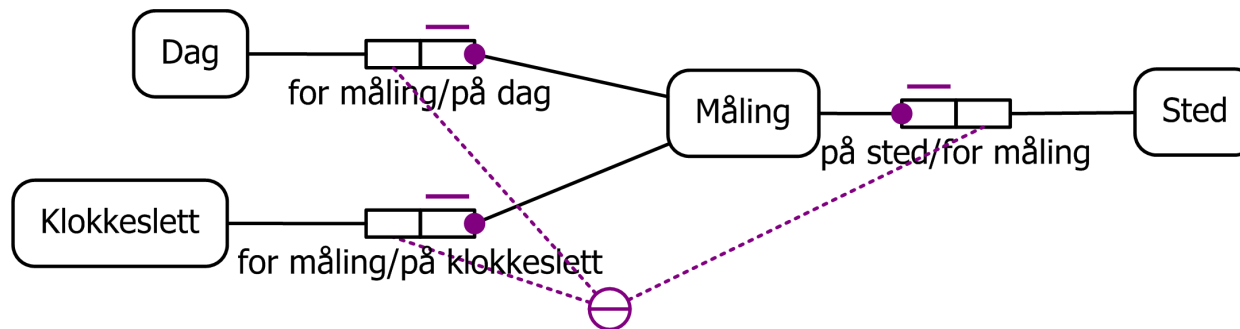


- Relasjonen **Måling** får et attributt **Temperatur_er_resultat_av**
- Attributtet **Temperatur_er_resultat_av** blir fremmednøkkel til relasjonen **Temperatur**
- Tilsvarende får **Måling** attributtet **Person_foretok** som er fremmednøkkel til relasjonen **Person**

Måling(Dato, TimeMinuttSekund, Stedsnavn, Temperatur_er_resultat_av, Person_foretok)

Fremmednøkler: Måling(Temperatur_er_resultat_av) → Temperatur(°C)
Måling(Person_foretok) → Person(PersonId)

Begrepsdannelse og fremmednøkler



De binære faktatypene som inngår i begrepsdannelser, gir også opphav til fremmednøkler:

Måling(Dato, TimeMinuttSekund, Stedsnavn, Temperatur_er_resultat_av, Person_foretok)

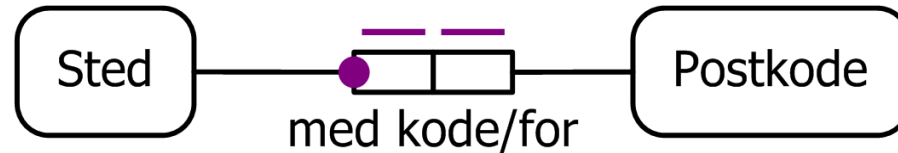
Fremmednøkler (i tillegg til fremmednøklerne på forrige side):

Måling(Dato) → Dag(Dato)

Måling(TimeMinuttSekund) → Klokkeselett(TimeMinuttSekund)

Måling(Stedsnavn) → Sted(Stedsnavn)

Gruppering av 1:1-faktatyper



- Med to entydige roller kan begge velges som gruppererrolle
- Hvis én av rollene er påkrevd (som tilfellet er for **med kode**), bør denne velges
- På grunn av entydighetsskranken over referanserollen er fremmednøkkelen **Postkode_for** entydig. I dette tilfellet blir **Postkode_for** derfor en kandidatnøkkel for **Sted**.
(Vi setter i det videre to streker under primærnøkkelen og én strek under den andre kandidatnøkkelen i **Sted**.)

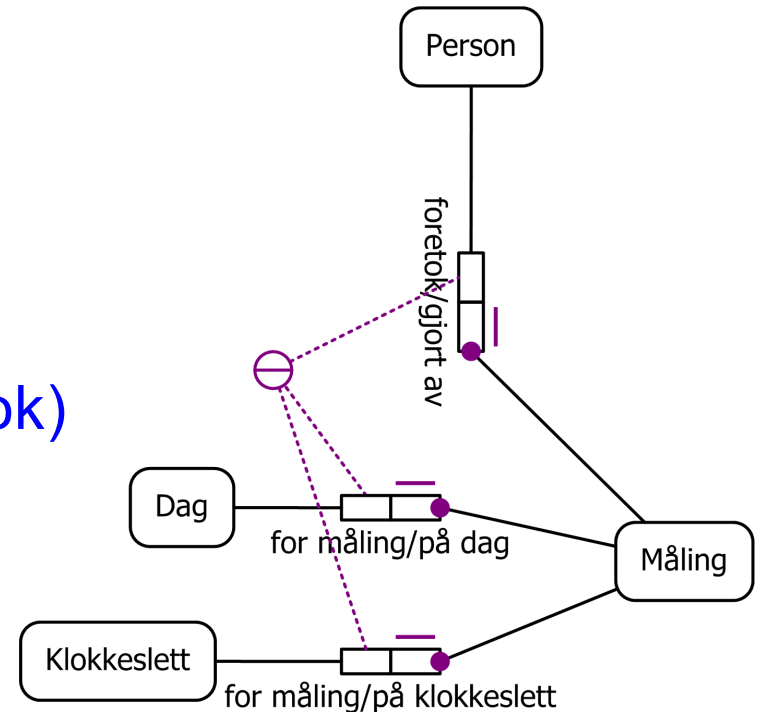
Sted(Stedsnavn, Postkode_for)

Fremmednøkler: Sted(Postkode_for) → Postkode(Zip)

Eksterne entydigheter og kandidatnøkler

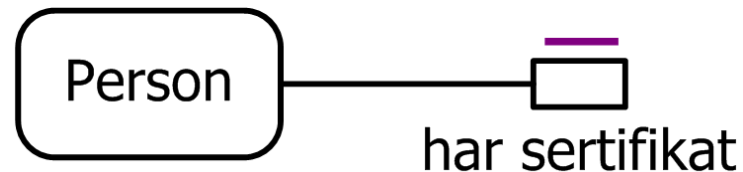
Eksterne entydigheter som ikke inngår i referansemåtene, gir også opphav til kandidatnøkler:

I **Måling** er
(Dato, TimeMinuttSekund, Person_foretok)
en kandidatnøkkel.



Måling(Dato, TimeMinuttSekund, Stedsnavn, Temperatur_er_resultat_av, Person_foretok)

Gruppering av unære faktatyper

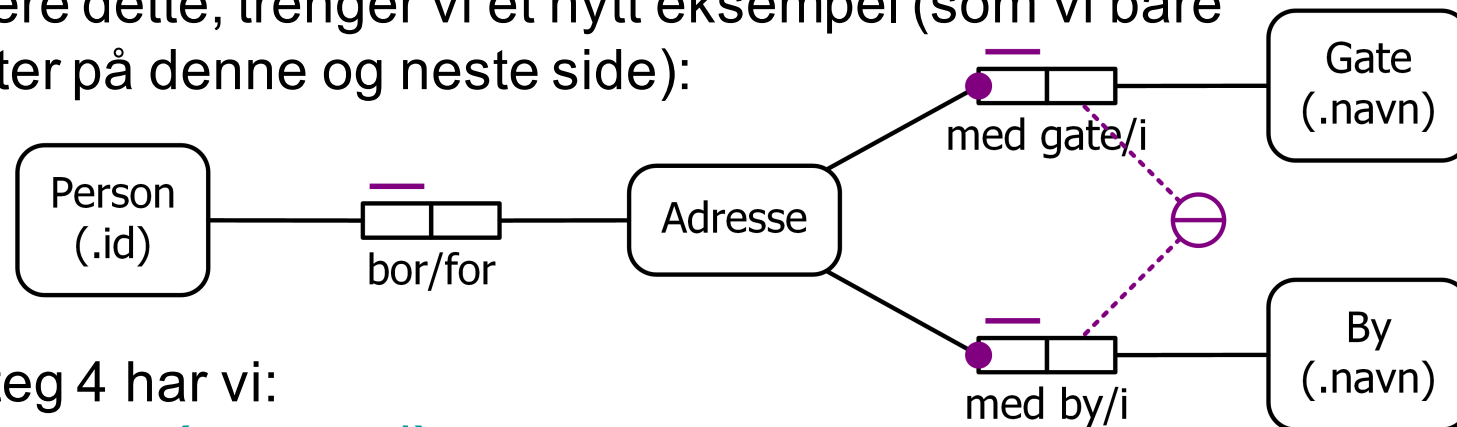


- Relasjonen **Person** får et boolsk attributt **Har_sertifikat**:
Person(PersonId, [Navn_for], Har_sertifikat)
- Nullverdier er aldri tillatt for boolske attributter som stammer fra unære faktatyper
- Rollen i en unær faktatype kan aldri være påkrevd
- Vi kan klare oss uten unære faktatyper:



Antall attributter i fremmednøkler

- Merk at en fremmednøkkel kan inneholde mer enn ett attributt. For å illustrere dette, trenger vi et nytt eksempel (som vi bare benytter på denne og neste side):



- Før steg 4 har vi:

Person(PersonId)

Gate(Gatenavn)

By(Bynavn)

Adresse(Gatenavn, Bynavn)

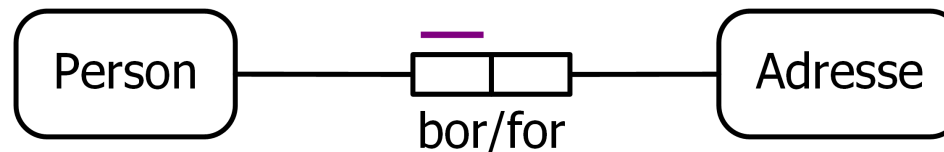
Fremmednøkler:

Adresse(Gatenavn) → Gate(Gatenavn)

Adresse(Bynavn) → By(Bynavn)

Gruppering av den binære faktatypen

- I steg 4 skal vi utvide relasjonen Person med attributter som utgjør en fremmednøkkel til Adresse, i henhold til den binære faktatypen



- Etter steg 4 er Person endret til
Person(PersonId, Gatnavn_for, Bynavn_for)
Fremmednøkkel:
Person(Gatnavn_for, Bynavn_for) → Adresse(Gatnavn, Bynavn)
- Merk at en fremmednøkkel alltid peker ut primærnøkkelen i den refererte relasjonen.

Fremmednøkler

Person(PersonId, [Navn_for], Har_sertifikat)
Dag(Dato)
Klokkeslett(TimeMinuttSekund)
Postkode(Zip)
Sted(Stedsnavn, Postkode_for)
Måling(Dato, TimeMinuttSekund, Stedsnavn, Temperatur_er_resultat_av, Person_foretok)
Temperatur(°C)

Fremmednøkklene er altså:

Sted(Postkode_for) → Postkode(Zip)
Måling(Dato) → Dag(Dato)
Måling(TimeMinuttSekund) → Klokkeslett(TimeMinuttSekund)
Måling(Stedsnavn) → Sted(Stedsnavn)
Måling(Temperatur_er_resultat_av) → Temperatur(°C)
Måling(Person_foretok) → Person(PersonId)

Steg 5: Overføring av skranker

5. Overfør resten av skrankene til relasjonsskjemaet

Endel skranker, som ulikhet, delmengde mm., har vi ikke sagt noe om ennå. Vi kommer tilbake til disse i en senere forelesning.

Steg 6: Undertrykkelse

6. Bestem hvilke referanserelasjoner som skal fjernes

Person(PersonId, [Navn_for], Har_sertifikat)

Dag(Dato)

Klokkeslett(TimeMinuttSekund)

Postkode(Zip)

Sted(Stedsnavn, Postkode_for)

Måling(Dato, TimeMinuttSekund, Stedsnavn, Temperatur_er_resultat_av, Person_foretok)

Temperatur(°C)

- Ikke alle relasjonene over er like interessante. **Dag**, **Klokkeslett** og **Temperatur** inneholder henholdsvis datoer, tidspunkter og temperaturer - men alle datoer, tidspunkter og temperaturer som er av interesse, forekommer også i **Måling**, så er det da noe poeng i å ha egne relasjoner/tabeller for disse?
- Slike relasjoner kan fjernes fra skjemaet (**undertrykkes**). Vi kommer tilbake til hvilke relasjoner som kan undertrykkes i en senere forelesning.